Implementing the Modified LH Algorithm

I. Krohn, S. Moltzahn, J. Rosenmüller, P. Sudhölter, and H.-M. Wallmeier

Institute of Mathematical Economics
University of Bielefeld
D-4800 Bielefeld, Germany

Transmitted by John Casti

ABSTRACT

The modified Lemke-Howson algorithm is a constructive procedure which enables us to compute equilibrium points of a bimatrix game. The algorithm as described by one of the authors is based on the original version invented by Lemke and Howson. However, it differs from this version with respect to several features. It works directly with the matrices defining the bimatrix game A and B. It has an easy and very direct geometrical interpretation; hence for small games we can follow the development of the algorithm geometrically. Finally, instead of being bilinear, the algorithm behaves rather like a piecewise linear program. This presentation closes a gap: although the algorithm has been described geometrically (and with a flow diagram), there has been no constructive procedure that can be implemented on a computer. This is provided by the present paper. We give all necessary proofs and computations in order to establish the following facts: There are *two* tableaus accompanying the proceeding of the algorithm. As the algorithm changes, moving alternatingly in the simplices of mixed strategies, so does the computational procedure alternatingly dealing with the two different tableaus. Each tableau contains six regions, depending on the various sequences of transitions the procedure has to perform. While this all is in marked difference to linear programming, there is also consolation: The well-known rectangle rule of linear programming can be modified easily (that is, there is a family of rectangle rules) so that changing the tableau alternatingly amounts to applying the appropriate rectangle rule. Thus, there is also close similarity to the familiar LP procedure. Thus, a complete description of the modified LH algorithm is provided that can immediately be implemented on any computer. In particular, we supply an APL program that, e.g., can be run on an IBM® PC.

655 Avenue of the Americas, New York, NY 10010 0096-3003/91/$03.50

1. INTRODUCTION

Let $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, n\}$. A *bimatrix game* (in mixed strategies) is a quadruple

$$\Gamma = (X, Y, A, B) \tag{1.1}$$

such that $A = (a_{ij})_{i \in I,\, j \in J}$ and $B = (b_{ij})_{i \in I,\, j \in J}$ are $m \times n$ matrices and

$$X = \left\{ x \in \mathbb{R}^m \,\middle|\, x = (x_1, \ldots, x_m) \geqslant 0,\ \sum_{i \in I}^{m} x_i = 1 \right\}, \tag{1.2}$$

$$Y = \left\{ y \in \mathbb{R}^n \,\middle|\, y = (y_1, \ldots, y_n) \geqslant 0,\ \sum_{j \in J}^{n} y_j = 1 \right\} \tag{1.3}$$

are the (mixed) strategies of players 1 and 2. If player 1 chooses $x \in X$ and player 2 chooses $y \in Y$, then *payoffs* are defined by

$$xAy = \sum_{i \in I} \sum_{j \in J} x_i a_{ij} y_j$$

for player 1, and xBy for player 2.

A pair $(\bar{x}, \bar{y}) \in X \times Y$ is said to be an *equilibrium point* if

$$\bar{x}A\bar{y} \geqslant xA\bar{y} \qquad (x \in X)$$

and

$$\bar{x}B\bar{y} \geqslant \bar{x}By \qquad (y \in Y);$$

thus, in equilibrium, no player has an incentive to deviate, for his payoff cannot be improved upon. If Γ is a *zero-sum* game (i.e., $B = -A$), then an equilibrium consists of a pair of optimal strategies and vice versa.

The *Lemke-Howson algorithm*, as devised in [7], is a procedure that (for *nondegenerate* games) yields an equilibrium point within finitely many steps. The procedure works by transforming the bimatrix game into a bilinear program, whereafter the algorithm, starting with an unbounded edge, proceeds by moving along a certain system of polyhedral edges of dimension 1 to search for an equilibrium point. An implementation of this version of the

LH algorithm, in the sense that the geometrical behavior of the algorithm is represented by a sequence of tableaus, to be computed consecutively and leading to a numerical evaluation of an equilibrium point, has been presented by Parthasarathy and Raghavan [8]; however, a formal proof (and an established computer program) for a neatly working version of the algorithm on a modern computer is lacking.

The LH algorithm also yields some insight into the structure of equilibria. It shows that the number of equilibrium points (for nondegenerate games) is odd. It is also known that not every equilibrium point can necessarily be reached in any case; even if the initial "unbounded edge" can be changed, there are equilibria not to be reached by the LH algorithm (for further discussion we refer to Aggarval [1], Bastian [2], Eaves [5], Lemke [6], Parthasarathy and Raghavan [8], Shapley [9] and Todd [12, 13]).

An alternative version of the algorithm (the *modified LH algorithm*) has been presented by Rosenmüller [11, Chapter I, Section 1]. This version works directly with the matrices, with A and B constituting the bimatrix game. The algorithm is not bilinear but rather piecewise linear: it works effectively in the simplices X and Y, alternately performing steps in each of them. There is a flow diagram established in [11], which, however, requires the computation of solutions of certain linear equations after each step and hence is not in the spirit of traditional linear programming. In practice the procedure suggested by the flow diagram is rather slow, and the capacity of most computers is not sufficient, even for small problems.

As the procedure is not a standard optimization problem, it is not clear exactly how to define a sequence of tableaus corresponding to the geometrical movement of the LH algorithm as presented in [11]. This is the goal of the present paper. We suggest the correct parametrization of edges of certain subpolyhedra of the simplices of mixed strategies X and Y. Using this parametrization, we define a pair of tableaus (corresponding to the alternating behavior of the modified LH algorithm) such that alternately executing the rectangle rule in each of the tableaus actually yields an equilibrium point. The procedure can thus be implemented on a computer, and for the sake of completeness we include an APL version of such a program.

Let $A_{i\cdot}, A_{\cdot j}$ denote the ith row and jth column of the matrix A respectively. Introduce the convex polyhedra

$$K_i = \{y \in Y \mid A_{i\cdot}y \geqslant A_{k\cdot}y \ (k \in I)\}, \qquad (i \in I)$$

$$L_j = \{x \in X \mid xB_{\cdot j} \geqslant xB_{\cdot l} \ (l \in J)\}, \qquad (j \in J) \tag{1.4}$$

as well as

$$K_T = \bigcap_{i \in T} K_i \qquad T \subseteq I, \quad T \neq \varnothing,$$
$$L_R = \bigcap_{j \in R} L_j \qquad R \subseteq J, \quad R \neq \varnothing. \tag{1.5}$$

Here, K_i denotes the mixed strategies of player 2 against which the (pure) strategy $i \in I$ of player 1 is the best reply. It is not hard to see that $(\bar{x}, \bar{y})$ is an equilibrium point of Γ if and only if

$$\bar{y} \in K_{\{i \mid \bar{x}_i > 0\}} \quad \text{and} \quad \bar{x} \in L_{\{j \mid \bar{y}_j > 0\}}.$$

Thus, in equilibrium, the positive coordinates of $\bar{x}$ and the polyhedra K_i containing $\bar{y}$ correspond to each other (in fact uniquely if nondegeneracy prevails)—this is of course an analogue to the familiar "optimality condition" of LP theory. We are thus motivated to introduce polyhedra

$$H_{T,U} = K_T \cap \left\{ y \in Y \middle| y_j = 0 \ (j \in U) \right\},$$
$$G_{R,V} = L_R \cap \left\{ x \in X \middle| x_i = 0 \ (i \in V) \right\}. \tag{1.6}$$

The game is called *nondegenerate* if

$$\dim H_{T,U} = n - |T| - |U|, \quad \dim G_{R,V} = m - |R| - |V|$$
$$\text{for} \quad H_{T,U} \neq \varnothing \neq G_{R,V} \tag{1.7}$$

(cf. Definition 1.11, Section 1, Chapter 1 of [9]). *We shall assume that the game we are dealing with is nondegenerate.*

In this case we have the following characterization of equilibrium points:

Let $(\bar{x}, \bar{y}) \in X \times Y$, and put $T = \{i \mid \bar{x}_i > 0\} \subseteq I$ and $R = \{j \mid \bar{y}_j > 0\} \subseteq J$. Then $(\bar{y}, \bar{x})$ is an equilibrium point if and only if $|T| = |R|$ and $\{(\bar{y}, \bar{x})\} = H_{T,R^c} \times G_{R,T^c}$. (1.8)

For the details, see [11], and in particular Corollary 1.13 in Section 1 of Chapter 1.

The statement formalized in (1.8) can be interpreted geometrically as follows: the simplices X and Y of mixed strategies are *decomposed* by the polyhedra L_j $(j \in J)$ and K_i $(i \in I)$ respectively. Among the subfaces of such polyhedra we distinguish *vertices* $H_{T,U}$, $|T|+|U|=n$, and edges $H_{T,U}$, $|T|+|U|=n-1$ (for some $K_i \subseteq Y$; the situation is analogously described in X). A vertex $H_{T,U}=\{\bar{y}\}$ has "labels" assigned to it by the polyhedra it is adjacent to (i.e., labels $i \in T$ with $\bar{y} \in K_i$) and by the positive coordinates of $\bar{y}$ (i.e., $\bar{y}_j > 0$ for $j \in U^c$). If $(\bar{x}, \bar{y})$ is an equilibrium point, then the labels of $\{\bar{x}\} = G_{R,V}$ and $\{\bar{y}\} = H_{T,U}$ correspond to each other in a unique way.

EXAMPLE 1.1. Consider the matrices

$$A = \begin{pmatrix} 5 & 3 & -4 & -1 \\ -6 & -3 & 5 & 3 \end{pmatrix}$$

and

$$B = \begin{pmatrix} -1 & -4 & 7 & 11 \\ 3 & 4 & -9 & -19 \end{pmatrix}.$$

then the sketch in Figure 1 illustrates the decomposition of X into polyhedra L_1, L_2, L_3, L_4 and the decomposition of Y into polyhedra K_1, K_2. An equilibrium point is given by

$$\bar{x} = \left(\tfrac{3}{5}, \tfrac{2}{5}\right), \qquad \{\bar{x}\} = G_{\{1,3\},\varnothing} = G_{\{1,3\},\{1,2\}^c},$$

$$\bar{y} = \left(\tfrac{9}{20}, 0, \tfrac{11}{20}, 0\right), \qquad \{\bar{y}\} = H_{\{1,2\},\{2,4\}} = H_{\{1,2\},\{1,3\}^c},$$

where the indices ("labels") are matched in the appropriate way: $\bar{x}$ has positive coordinates 1, 2 and $\bar{y} \in K_1 \cap K_2$; analogously $\bar{y}_1 > 0$, $\bar{y}_3 > 0$, while $\bar{x} \in L_1 \cap L_3$.

The modified LH algorithm is explained in detail in Chapter 1, Section 1 of Rosenmüller [11]; see also [10] for the n-person game version (Wilson [14] describes the "multilinear" n-person version of the "original" LH algorithm). We would like to assume the reader is slightly familiar with the presentation in [11].

For our present purpose we shall describe the modified LH algorithm with the aid of Example 1.1 as follows: Use e^i to denote the ith unit vector.

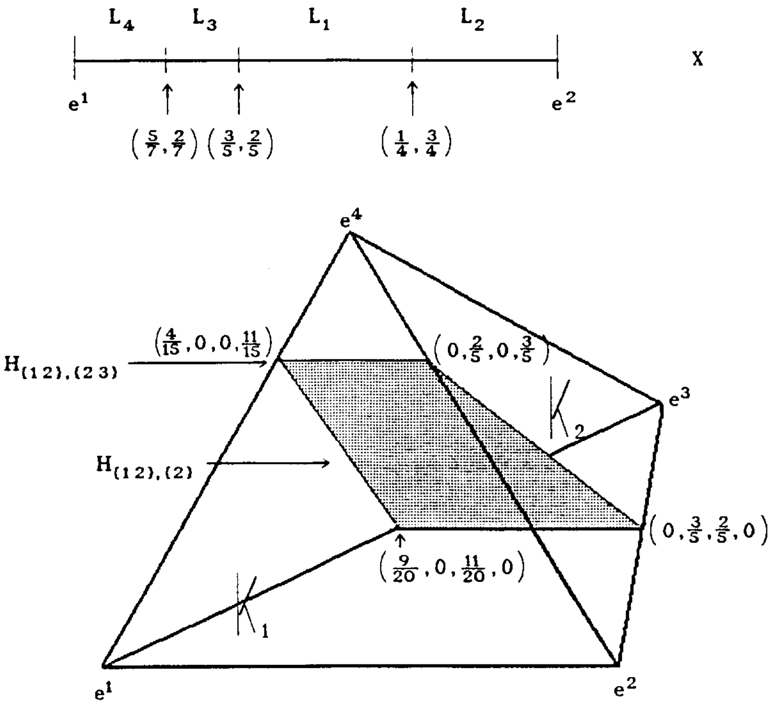


FIG. 1.

The algorithm starts with a vertex, say

$$\{e^4\} = H_{\{2\},\{123\}} \subseteq Y,$$

in Y. As $e^4 \in K_2$, we move to simplex X and choose

$$\{e^2\} = G_{\{2\},\{1\}}$$

as the first vertex in X. Now, $e^2 \in L_2$ means that, in Y, we should admit positive 2nd coordinates, i.e. move "towards" $e^2 \in Y$. That is, we delete index 2 from the labels describing $\{e^4\} \in Y$, thus moving along the edge

$$H_{\{2\},\{1,3\}},$$

the endpoint of which is

$$y^1 = \left(0, \tfrac{2}{5}, 0, \tfrac{3}{5}\right),$$

defining a vertex

$$\{y^1\} = H_{\{1,2\},\{1,3\}}.$$

Here the new label $i = 1$ has appeared; thus in X, we leave e^2 along the edge

$$G_{\{2\},\varnothing},$$

arriving at $x^1 = (\frac{1}{4}, \frac{3}{4})$, where

$$\{x^1\} = G_{\{1,2\},\varnothing}.$$

Hence the next edge in Y is $H_{\{1,2\},\{3\}}$, which leads to $y^2 = (\frac{4}{15}, 0, 0, \frac{11}{15})$. We have

$$\{y^2\} = H_{\{1,2\},\{2,3\}}.$$

The next steps are along the edge $G_{\{1\},\varnothing}$ towards $x^2 = (\frac{3}{5}, \frac{2}{5})$, $\{x^2\} = G_{\{13\},\varnothing}$, and along $H_{\{1,2\},\{2\}}$ towards $y^3 = (\frac{9}{20}, 0, \frac{11}{20}, 0)$. Now $\{y^3\} = H_{\{1,2\},\{2,4\}}$, and all labels match in the required manner, as we have explained above. Thus we have reached the equilibrium point.

The main purpose of this paper is to develop the computational procedure that accompanies the geometrical picture we have just studied. To this end we shall explain what kind of "movement along an edge" we should adopt for the rigorous mathematical representation. In other words, we shall define the *canonical parametrization* of edges, depending, however, on what kind of movement along an edge we have in mind. For (working in Y), according to whether we leave a polyhedron K_i (i.e. delete a label $i \in H_{T,U}$) or whether we leave a subface of Y (i.e. delete an index $j \in H_{T,U}$), there are two ways of departing from a vertex in order to move along an edge. Similarly, there are two ways of arriving at a vertex after having traveled along an edge. This yields different types of journey, and the canonical parametrization of this journey along an edge must be chosen accordingly. The appropriate choice is then reflected by the appropriate definition of the two tableaus corresponding to a pair of edges, each one located in a simplex X or Y respectively.

The development of our presentation is as follows. In the rest of Section 1 we shall again treat the four ways of traveling along an edge (the detailed discussion has been performed already in [10]). We shall then extensively discuss the case which is most typical for the modified LH algorithm. The other three cases will not be treated in detail. Hence, Section 2 is devoted to developing the canonical parametrization for case 1a of Figure 2 and to explaining the introductory data of the tableau corresponding to a vertex. In Section 3 we define the tableau (actually a pair of tableaus) and introduce the well-known rectangle procedure (which, though in structure resembling the one used in linear programming procedures, is quite different in its detailed appearance). We then prove that the rectangle rule, applied to the tableaus, accomplishes the journey between two edges; again the proof is presented in detail for just one particular case, whereas the other cases are treated superficially. Section 4 then collects the pieces: we present a detailed instruction for using the algorithm. That is, given the matrices A and B, it is explained how to set up the initial tableaus and perform the necessary steps in order to reach a final tableau. This eventually yields a pair of vectors constituting an equilibrium point of the game $\Gamma = (X, Y, A, B)$. Finally, in Section 5, for the sake of completeness, we include a computer program that actually performs the necessary computations. The program has been written in APL and was run on the IBM 6150 RT computer (IBM 6150 is a trademark of International Business Machines Cooperations). However, it can be implemented on any personal computer endowed with APL.

Let us finish this section by introducing the necessary notational conventions.

The matrices A and B are fixed throughout our presentation. In order to avoid indices (coordinates) $m+1, n+1$, we put

$$\mathbf{I} = \{1, \ldots, m, \square\} = I \cup \{\square\},$$

$$\mathbf{J} = \{1, \ldots, n, *\} = J \cup \{*\},$$

and similarly, for $T \subseteq I$, $U \subseteq J$,

$$\mathbf{T} = T \cup \{\square\}, \qquad \mathbf{U} = U \cup \{*\}.$$

Next, vectors $x \in \mathbb{R}^m$ are also repeated as functions $x : I \to \mathbb{R}$. Thus we denote the restriction of x onto $T \subseteq I$ by x_T; this is of course to be identified with the vector $x_T = (x_i)_{i \in T} \in \mathbb{R}^T$. For convenience we write

$$x_{-T} := x_{T^c} = x_{I-T},$$

so that for $z = (x_1, \ldots, x_m, \lambda) \in \mathbb{R}^{\mathbf{I}}$ we have e.g.

$$z_{-\mathbf{T}} : \mathbf{I} - \mathbf{T} \to \mathbb{R}$$

$$z_{-\mathbf{T}} = (x_i)_{i \notin T}.$$

Frequently singletons $\{i\} \subseteq I$ and their elements are identified; thus

$$x_i = x_{\{i\}} \quad \text{and} \quad x_{-i} = x_{-\{i\}} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m)$$

for $x \in \mathbb{R}^m$. In this context, "+" is used for "$\cup$" in the case of a disjoint union, e.g.,

$$T + i = T \cup \{i\} \qquad (\text{for} \quad i \notin T \subseteq I),$$

$$T + \square - i_0 = (T \cup \{\square\}) - \{i_0\} \qquad (\text{for} \quad i_0 \in T \subseteq I),$$

etc.

The disjoint union of subsets of I, say, is accomplished by the formation of a direct sum of functions (vectors) defined on these subsets. For example, if $T', T'' \subseteq I$, $T' \cap T'' = \varnothing$, and $z' : T' \to \mathbb{R}$, $z'' : T'' \to \mathbb{R}$, then $z = z' \oplus z''$ denotes the function on $T = T' + T''$ $(= T' \cup T'')$ defined by

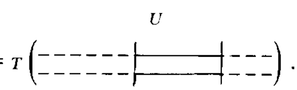
$$z : T' + T'' \to \mathbb{R},$$

$$z_i = z'_i \ (i \in T'), \qquad z_i = z''_i \ (i \in T''),$$

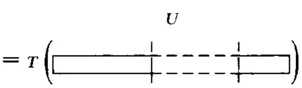
or (in less precise notation) the vector

$$z = (z', z'').$$

An analogous notation is employed for matrices. For example, the matrix A can be seen as a mapping $A : I \times J \to \mathbb{R}$, and for $T \subseteq I$, $U \subseteq J$ we denote by A_T^U the restriction on $T \times U$, which is represented by the matrix

$A_T^U =$ 

Similarly

$A_T^{-U} =$ 

We write $A_T := A^J_T$; however, the i'th row of A is $A_{i\cdot}$ and the j'th column is $A_{\cdot j}$; thus

$$A_{i\cdot} = A^J_{\{i\}} = A^J_i,$$

but A_i is avoided.

Next let $e = (1, \ldots, 1)$ (used for $e \in \mathbb{R}^m$ and $e \in \mathbb{R}^n$). Write

$$\mathfrak{A} = \begin{pmatrix} & & & -1 \\ & A & & \vdots \\ & & & -1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}$$

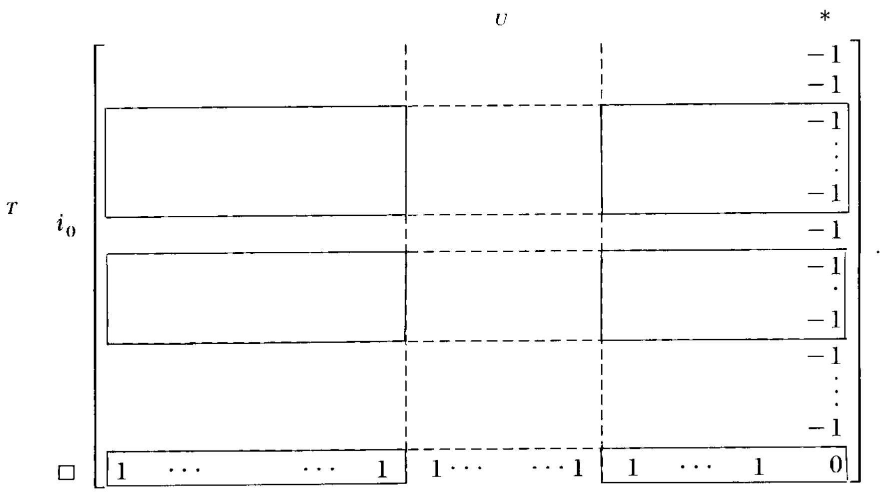
and

$$\mathfrak{B} = \begin{pmatrix} & & & 1 \\ & B & & \vdots \\ & & & 1 \\ -1 & \cdots & -1 & 0 \end{pmatrix}.$$

Thus, $\mathfrak{A} : \mathbf{I} \times \mathbf{J} \to \mathbb{R}$, and if $i_0 \in T \subseteq I \subseteq \mathbf{I}$ and $U \subseteq J \subseteq \mathbf{J}$, then it is seen that

$$\mathfrak{A}^{-U}_{\mathbf{T}-i_0} = \mathfrak{A}^{\mathbf{J}-U}_{T+\square-i_0} = \mathfrak{A}^{-U}_{-(T^c+i_0)}$$

is represented by



2. THE CANONICAL PARAMETRIZATION

Let us focus our interest on the motion which the modified LH algorithm performs in Y. Basically, there are four types of transitions that occur when the algorithmic procedure leaves a vertex, moves along an edge, and reaches the next vertex—geometrically speaking. These transitions can be classified according to whether a subface of Y is being left (reached) or a polyhedron K_i is being left (reached) upon departure (arrival).

Again, the details are explained in [9, Chapter 1, Section 1]; hence, for our present purpose we merely illustrate the four types of transitions for the case that A and B are $n\times 3$, $n>3$, matrices; see Figure 2. Here, $H_{T,U}=\{\bar{y}\}\subseteq Y$ denotes the departure vertex, while the arrival vertex varies accordingly; e.g., in case 1a we have $H_{\hat{T},\hat{U}}=\{\hat{y}\}=H_{T-i_0+i_1,U}$, etc.

Let us start out with an extensive discussion of case 1a. We shall define a certain version of a parametrization of the edge $H_{T-i_0,U}$ joining $\bar{y}$ and $\hat{y}$, called the *canonical* one. This will suggest (at least partially) the form of the corresponding tableau and the way the tableau changes when the algorithm switches from $\bar{y}$ to $\hat{y}$.

To this end, let us now fix an extreme point or vertex

$$Y\supseteq H_{T,U}=\{\bar{y}\}$$

such that $|T|\geqslant 2$ and $|T|+|U|=n$; put

$$\bar{\lambda}:=A_{i\cdot}\bar{y}\qquad (i\in T)$$

and define, for some fixed $i_0\in T$,

$$L_{T-i_0}^{-U}:=\left\{\mu=(\gamma,\nu)\in\mathbb{R}^{-U}\times\mathbb{R}\,\middle|\,\mathfrak{A}_{T-i_0}^{-U}\mu=0\right\}.$$

Then we have

LEMMA 2.1.

(1) *$L_{T-i_0}^{-U}$ is a linear subspace of $\mathbb{R}^{-U}\times\mathbb{R}$ with dimension* 1.
(2) *$\mathfrak{A}_{i_0\cdot}^{U}\mu\neq 0$ for all $\mu=(\gamma,\nu)\in L_{T-i_0}^{-U}$ with $\gamma\neq 0$.*

PROOF. Follows immediately, since the game is nondegenerate.

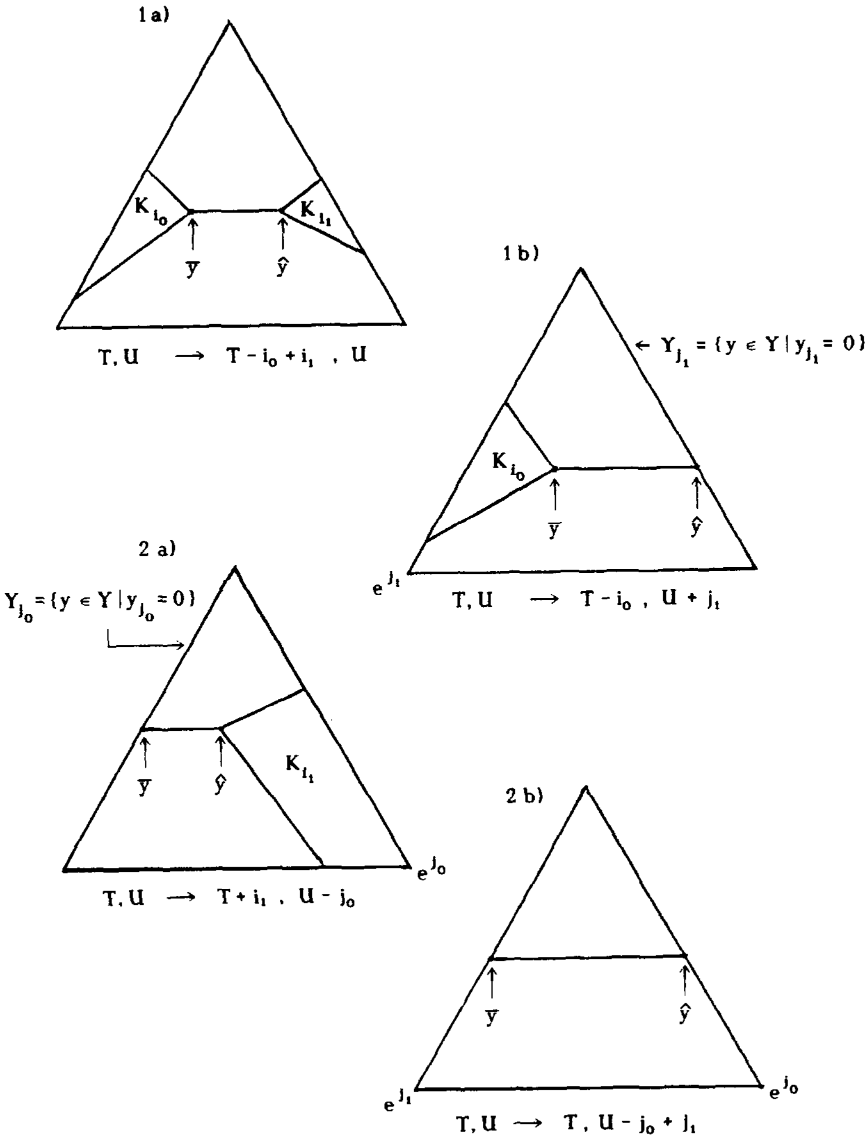

1 a)
K_{i_0}
K_{i_1}
ȳ
ŷ
T, U ⟶ T - i_0 + i_1 , U
1 b)
← Y_{j_1} = {y ∈ Y | y_{j_1} = 0}
K_{i_0}
ȳ
ŷ
e^{j_1}
T, U ⟶ T - i_0 , U + j_1
2 a)
Y_{j_0} = {y ∈ Y | y_{j_0} = 0}
ȳ
ŷ
K_{i_1}
e^{j_0}
T, U ⟶ T + i_1 , U - j_0
2 b)
ȳ
ŷ
e^{j_1}
e^{j_0}
T, U ⟶ T , U - j_0 + j_1

FIG. 2.

DEFINITION 2.2. For $i_0 \in T$ let

$$\bar{\mu}^{i_0} = \left(\bar{\gamma}^{i_0}, \bar{\nu}^{i_0}\right) \in L_{T-i_0}^{-U}$$

be defined by the requirement that

$$\mathfrak{A}_{i_0}^{-U} \bar{\mu}^{i_0} = 1 \tag{2.1}$$

hold true.

Let us observe that the quantities of generic types $\mu = (\gamma, \nu)$, as considered so far, can be naturally extended to vectors of $\mathbb{R}^n \times \mathbb{R}$ by adding zero coordinates for all $i \in U$. More precisely,

$$L_{T-i_0}^{-U} \oplus O_U \subseteq \mathbb{R}^n \times \mathbb{R}$$

is likewise a linear subspace of $\mathbb{R}^n \times \mathbb{R}$ with dimension 1, and $\bar{\mu}^{i_0} \oplus O_U$ is a distinctive element of this subspace. Accordingly,

$$(\bar{y}, \bar{\lambda}) + \left(L_{T-i_0}^{-U} \oplus O_U\right)$$

is an affine subspace of $\mathbb{R}^n \times \mathbb{R}$ with distinctive elements $(\bar{y}, \bar{\lambda})$ and $(\bar{y}, \bar{\lambda}) + (\bar{\mu}^{i_0} \oplus O_U)$. In view of Definition 2.2, we have obviously

$$\mathfrak{A}_{\mathrm{T}}(\bar{y}, \bar{\lambda}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{\mathrm{T}}, \tag{2.2}$$

$$\mathfrak{A}_{\mathrm{T}}\left((\bar{y}, \bar{\lambda}) + (\bar{\mu}^{i_0} \oplus O_U)\right) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i_0 \\ \\ \\ \\ \\ \end{matrix} \in \mathbb{R}^{\mathrm{T}}. \tag{2.3}$$

If we consider the projection of $\mathbb{R}^n \times \mathbb{R}$ onto $\mathbb{R}^n$, then the situation may be

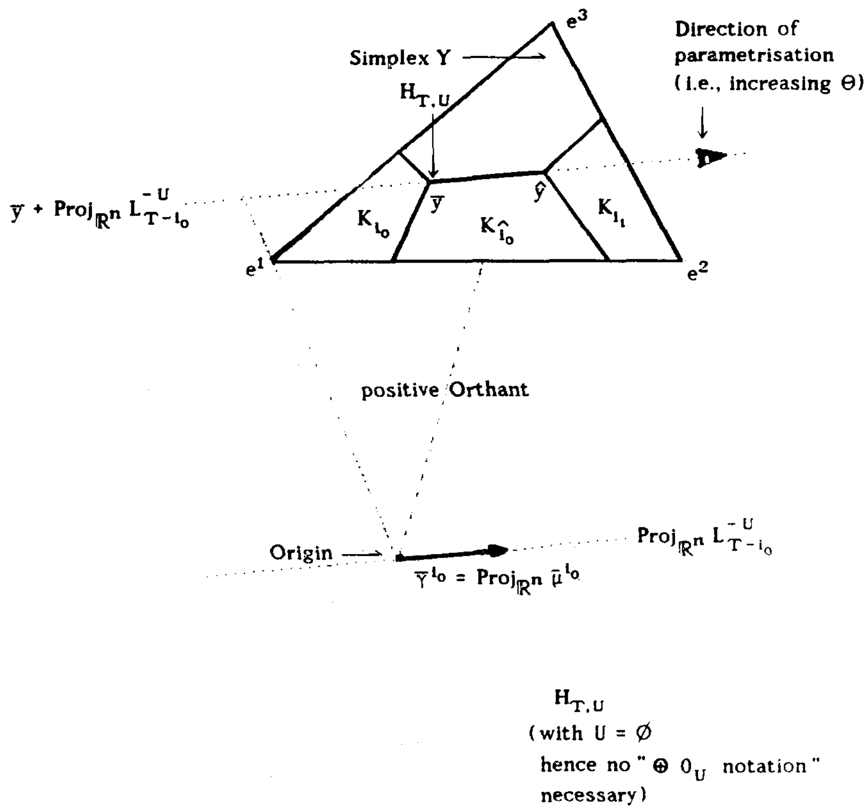


FIG. 3.

viewed in Figure 3—assuming that A and B have three columns. Also, Figure 3 represents the case in which $\bar{y}$ has positive coordinates—hence $U=\varnothing$.

DEFINITION 2.3. The *canonical parametrization* of $(\bar{y},\bar{\lambda})+(L_{T-i_0}^{-U}\oplus O_U)$ is the mapping

$$\theta \to (\bar{y},\bar{\lambda}) - \theta(\bar{\mu}^{i_0}\oplus O_U), \qquad \mathbb{R} \to \mathbb{R}^n \times \mathbb{R}. \tag{2.4}$$

We write $(y^\theta, \lambda^\theta) := (\bar{y}, \bar{\lambda}) - \theta(\bar{\mu}^{i_0} \oplus O_U)$ for $\theta \in \mathbb{R}$. (Actually, an additional index i_0 would be appropriate, but will be omitted for the sake of not overburdening our notation.)

THEOREM 2.4. *Let $\theta \to (y^\theta, \lambda^\theta)$ be the canonical parametrization described in Definition 2.3.*

(1) *There is $\bar{\theta}^{i_0} > 0$ such that*

$$H_{T-i_0,U} = \{y^\theta \mid 0 \leq \theta \leq \bar{\theta}^{i_0}\}.$$

(2) *$\bar{\theta}^{i_0}$ is explicitly computed by*

$$\bar{\theta}^{i_0} = \min\left\{ \frac{\bar{\lambda} - A_{i\cdot}\bar{y}}{\bar{\nu}^{i_0} - A_{i\cdot}^{-U}\bar{\gamma}^{i_0}} \,\middle|\, i \in T^c,\ \bar{\nu}^{i_0} > A_{i\cdot}^{-U}\bar{\gamma}^{i_0} \right\}$$

$$\wedge \min\left\{ \frac{\bar{y}_j}{\bar{\gamma}_j^{i_0}} \,\middle|\, j \in U^c,\ \bar{\gamma}_j^{i_0} > 0 \right\}.$$

(3) *For $i \in T - i_0$ and $i' \in T^c + i_0$, $0 < \theta < \bar{\theta}^{i_0}$, we have*

$$A_{i\cdot} y^\theta = \lambda^\theta > A_{i'\cdot} y^\theta$$

(4) *$y^{\bar{\theta}^{i_0}} =: \hat{y}$ is the second vertex (apart from $\bar{y}$) adjacent to the edge $H_{T-i_0,U}$.*

Note that in statement (2) the minimizer decides whether case 1a or case 1b is prevailing. That is, if the argmin is some $i_1 \in T^c$ and

$$\bar{\theta}^{i_0} = \frac{\bar{\lambda} - A_{i_1\cdot}\bar{y}}{\bar{\nu}^{i_0} - A_{i_1\cdot}^{-U}\bar{\gamma}^{i_0}},$$

then we are dealing with case 1a, etc.

PROOF OF THEOREM 2.4. In view of our previous construction, the affine one-dimensional subspace of

$$(\bar{y}, \bar{\lambda}) + \left(L_{T-i_0}^{-U} \oplus O_U\right),$$

which is parametrized by

$$\theta \to y^{\theta} \qquad (\theta \in \mathbb{R}),$$

contains the edge $H_{T-i_0,U}$. In particular, for $\theta = 0$ we have $y^0 = \bar{y} \in H_{T,U}$. In view of the defining property (2.1) of $\bar{\mu}^{i_0} = (\bar{\gamma}^{i_0}, \bar{\nu}^{i_0})$ we have clearly

$$A_{i_0\cdot}(\bar{\gamma}^{i_0} \oplus O_U) = \bar{\nu}^{i_0} + 1.$$

Also, exploring the $-$ sign in the canonical representation, we come up with

$$\begin{aligned} A_{i_0\cdot} y^{\theta} &= A_{i_0\cdot}\left[\bar{y} - \theta(\bar{\gamma}^{i_0} \oplus O_U)\right] \\ &= \bar{\lambda} - \theta\left[A_{i_0\cdot}(\bar{\gamma}^{i_0} \oplus O_U)\right] \\ &= \bar{\lambda} - \theta(\bar{\nu}^{i_0} + 1) \\ &= \lambda^{\theta} - \theta < \lambda^{\theta} = A_{i\cdot} y^{\theta} \end{aligned} \tag{2.5}$$

whenever $\theta > 0$ and $i \in T - i_0$. This implies

$$y^{\theta} \notin K_{i_0} \qquad (\theta > 0).$$

Hence, for sufficiently small $\theta > 0$ it turns out that $y^{\theta} \in H_{T-i_0,U}$ and $y^{\theta} \notin H_{T,U}$. By a compactness argument, statements (1), (3), and (4) of our theorem follow at once; it remains to show (2).

Now, clearly $y^{\theta} \in H_{T-i_0,U}$ for all θ that satisfy

$$y^{\theta} \geqslant 0, \qquad A_{i\cdot} y^{\theta} \leqslant \lambda^{\theta} \quad (i \in T^c), \tag{2.6}$$

and $\bar{\theta}^{i_0}$ is the smallest θ that violates one of the conditions (2.6), i.e., the smallest θ violating either

$$\bar{y}_j - \theta \bar{\gamma}_j^{i_0} > 0 \tag{2.7}$$

for some j with $j \in U^c$, $\bar{\gamma}_j^{i_0} > 0$, or

$$A_{i\cdot}\left[\bar{y} - \theta(\bar{\gamma}^{i_0} + O_U)\right] < \bar{\lambda} - \theta \bar{\nu}^{i_0}, \tag{2.8}$$

i.e.,

$$\theta\left(\bar{\nu}^{i_0} - A_i^{-U}\bar{\gamma}^{i_0}\right) < \bar{\lambda} - A_{i\cdot}\bar{y}$$

for some $i \in T^c$ with $\bar{\nu}^{i_0} > A_i^{-U}\bar{\gamma}^{i_0}$. Obviously, the θ we are looking for is the one given by (2). ■

So far our presentation has just been dealing with the departure vertex, which in cases 1a and 1b is obtained by sacrificing a condition $\bar{y} \in K_{i_0}$, i.e., by leaving K_{i_0}. Now, let us turn to the arrival, that is, as we want to treat case 1a, the entrance into some new K_{i_1}. In other words, let us consider the situation in which there is $i_1 \in T^c$ satisfying

$$\bar{\theta}^{i_0} = \frac{\bar{\lambda} - A_{i_1\cdot}\bar{y}}{\bar{\nu}^{i_0} - A_{i_1}^{-U}\bar{\gamma}^{i_0}} = \frac{\mathfrak{A}_{i_1\cdot}(\bar{y},\bar{\lambda})}{\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0}}. \tag{2.9}$$

This means that the vertex adjacent to $H_{T-i_0,U}$ (apart from $\bar{y}$) is

$$\left\{y^{\bar{\theta}^{i_0}}\right\} = H_{T-i_0+i_1,U}.$$

Let us write $\hat{y} := y^{\bar{\theta}^{i_0}}$.

Suppose now that, for $\hat{i}_0 \in T - i_0 + i_1$, we want to perform the same procedure as previously, yielding the canonical parametrization of $H_{T-i_0+i_1-\hat{i}_0,U}$. In this way we obtain the vector

$$\hat{\mu}^{\hat{i}_0} \in L_{T-i_0+i_1-\hat{i}_0}^{-U},$$

which, given $\hat{y}$, is defined by a requirement analogous to (2.1), i.e., by

$$\mathfrak{A}_{\hat{i}_0}^{-U}\hat{\mu}^{\hat{i}_0} = 1. \tag{2.10}$$

Define a quantity

$$\bar{c}_{i_1}^{i_0} := -\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0}. \tag{2.11}$$

Then it turns out that this quantity may be used to establish a direct relation between $\overline{\mu}^{i_0}$ and $\hat{\mu}^{\hat{\imath}_0}$ as follows:

COROLLARY 2.5. *Let* $i_1 \in T^c$ *satisfy* (2.9), *and suppose that* $\hat{\mu}^{\hat{\imath}_0} \in L^{-U}_{T-i_0+i_1-\hat{\imath}_0}$ *is given via* (10). *Then* [*using* (2.11)] *we have*

$$\hat{\mu}^{\hat{\imath}_0} = \overline{\mu}^{\hat{\imath}_0} - \frac{\overline{c}^{\hat{\imath}_0}_{i_1}}{\overline{c}^{i_0}_{i_1}} \overline{\mu}^{i_0} \qquad \textit{for} \quad \hat{\imath}_0 \neq i_1 \tag{2.12}$$

and

$$\hat{\mu}^{i_1} = -\frac{\overline{\mu}^{i_0}}{\overline{c}^{i_0}_{i_1}}. \tag{2.13}$$

PROOF. By definition of $\overline{\mu}^{i_0}$ we have

$$\mathfrak{A}^{-U}_{T-i_0} \overline{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{T-i_0};$$

hence

$$\mathfrak{A}^{-U}_{T-i_0+i_1} \overline{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathfrak{A}^{-U}_{i_1} \overline{\mu}^{i_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i_1 \\ \\ \\ \\ \end{matrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\overline{c}^{i_0}_{i_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i_1 \\ \\ \\ \\ \end{matrix} \in \mathbb{R}^{T-i_0+i_1}. \tag{2.14}$$

As (14) reads also

$$-\mathfrak{A}^{-U}_{T-i_0+i_1} \frac{\overline{\mu}^{i_0}}{\overline{c}^{i_0}_{i_1}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow i_1 \\ \\ \\ \\ \end{matrix} \in \mathbb{R}^{T-i_0+i_1},$$

it is seen that $-\bar{\mu}^{i_0}/\bar{c}^{i_0}_{i_1}$ satisfies the defining properties of $\hat{\mu}^{i_1}$—which proves (2.13).

Similarly, consider now the case $\hat{\imath}_0 \neq i_1$. We have

$$\mathfrak{A}^{-U}_{T-i_0+i_1-\hat{\imath}_0}\bar{\mu}^{i_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathfrak{A}^{-U}_{i_1}\bar{\mu}^{i_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\bar{c}^{i_0}_{i_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 \in \mathbb{R}^{T-i_0+i_1-\hat{\imath}_0}. \tag{2.15}$$

Next, the canonical parametrization at $\bar{y}$ with respect to $\hat{\imath}_0$ (which is an element of T) yields the quantity $\bar{\mu}^{\hat{\imath}_0}$, which is uniquely defined by the requirement that it satisfy

$$\mathfrak{A}^{-U}_{T}\bar{\mu}^{\hat{\imath}_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \hat{\imath}_0 \in \mathbb{R}^{T}. \tag{2.16}$$

Thus

$$\mathfrak{A}^{-U}_{T-i_0+i_1-\hat{\imath}_0}\bar{\mu}^{\hat{\imath}_0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathfrak{A}^{-U}_{i_1}\bar{\mu}^{\hat{\imath}_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\bar{c}^{\hat{\imath}_0}_{i_1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i_1 \in \mathbb{R}^{T-i_0+i_1-\hat{\imath}_0}. \tag{2.17}$$

Multiplying (2.15) with the appropriate factor and subtracting the result from (2.17), we come up with

$$\mathfrak{A}^{-U}_{T-i_0+i_1-\hat{\imath}_0}\left(\bar{\mu}^{\hat{\imath}_0} - \frac{\bar{c}^{\hat{\imath}_0}_{i_1}}{\bar{c}^{i_0}_{i_1}}\bar{\mu}^{i_0}\right) = 0 \in \mathbb{R}^{T-i_0+i_1-\hat{\imath}_0}. \tag{2.18}$$

Moreover, using (2.16) and Definition 2.2, we find

$$\mathfrak{A}_{\hat{\imath}_0}^{-U}\left(\bar{\mu}^{\hat{\imath}_0}-\frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}}\bar{\mu}^{i_0}\right)=1-\frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}}\mathfrak{A}_{\hat{\imath}_0}^{-U}\bar{\mu}^{i_0}$$

$$=1-0=1. \tag{2.19}$$

Concluding, we realize that (2.18) and (2.19) show $\bar{\mu}^{\hat{\imath}_0}-(\bar{c}_{i_1}^{\hat{\imath}_0}/\bar{c}_{i_1}^{i_0})\bar{\mu}^{i_0}$ to satisfy the conditions defining $\hat{\mu}^{\hat{\imath}_0}$ uniquely; this indeed verifies (2.12). ■

COROLLARY 2.6. *For $i_1 \in T^c$ let*

$$\overline{\Theta}_{i_1} := -\mathfrak{A}_{i_1\cdot}(\bar{y},\bar{\lambda}) = -A_{i_1\cdot}\bar{y}+\bar{\lambda}. \tag{2.20}$$

Then, for $j_1 \in U^c$,

$$\hat{y}_{j_1}=\bar{y}_{j_1}-\frac{\overline{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}\bar{\gamma}_{j_1}^{i_0} \tag{2.21}$$

and

$$\hat{\lambda}=\bar{\lambda}-\frac{\overline{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}\bar{\nu}^{i_0}. \tag{2.22}$$

PROOF. Indeed, since

$$\bar{c}_{i_1}^{i_0}=-\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0}=\bar{\nu}^{i_0}-A_{i_1}^{U^c}\bar{\gamma}^{i_0},$$

we can use (2.9) and (2.4) in order to obtain

$$\hat{y}=\bar{y}-\bar{\theta}^{i_0}(\bar{\gamma}^{i_0}\oplus O_U)=\bar{y}-\frac{\overline{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}(\bar{\gamma}^{i_0}\oplus O_U).$$ ■

Let us pause for some reflection. The development as presented so far describes the transition from the vertex $\bar{y}$ to the adjacent vertex $\hat{y}$ [assuming we consider case 1a, that is, $(T,U) \to (T - i_0 + i_1, U)$]. Equations (2.12) and (2.13) may reasonably be interpreted as an analogue to the well-known rectangle rule of linear programming.

Indeed, in order to compute $\hat{y}$ by means of $\bar{y}$ we need certain quantities $\bar{\gamma}_{\cdot}^{\cdot}, \bar{c}_{\cdot}^{\cdot}, \overline{\Theta}_{\cdot\cdot}$. Moreover, in order to compute the next adjacent vertex, we have to start with $\hat{y}$ and use the corresponding quantities, say $\hat{\gamma}_{\cdot}^{\cdot}$, $\hat{c}_{\cdot}^{\cdot}$, and $\hat{\Theta}_{\cdot\cdot}$. Hence, we have to find a computational rule for the transition of these quantities. To this end, we focus on Corollary 2.5, which indeed presents a version of the rectangle rule for a transformation of $\bar{\mu}$ to $\hat{\mu}$. This transformation in turn depends on the quantities $c_{\cdot}^{\cdot}$ as indicated by the result of Corollary 2.5. This means that we have to establish the rectangle rule for the quantities c and Θ as well. It seems advisable to combine all necessary quantities in what is usually called the *tableau* assigned to the vertex $\bar{y}$. This tableau should at least contain quantities $\bar{y}, \bar{\gamma}, \bar{c}, \overline{\Theta}$.

There is, however, a further obstacle: So far we have only discussed case 1a. There are four other cases, which conceivably would yield additional quantities to be represented in our tableau to be constructed. At this point, therefore, we prefer to present the tableau without further motivation. Rather, the quantities that will appear will be justified by further computation and transformational arguments following in the next sections.

3. THE TABLEAU

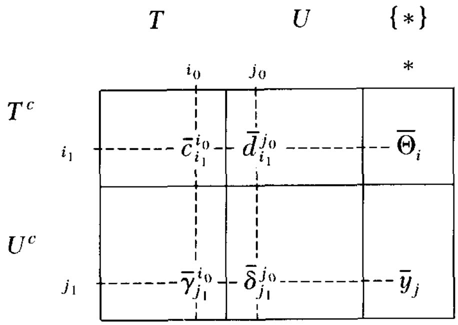
The peculiar pattern of the LH algorithm as presented in Section 1 asks for a slightly more complicated version of the tableau attached to a certain vertex $\{\bar{y}\} = H_{T,U}$. It should be noted that we still are discussing the situation in Y only. There is obviously a similar tableau attached to any vertex in the corresponding simplex X.

The tableau to be presented below contains six different regions, four of them corresponding to the defining subsets T and U and their complements respectively. According to what kind of transition (corresponding to cases 1a to 2b) is necessary, the rectangle rule will switch the coefficients, depending on the positions in the various regions of the tableau. Ideally, in order to compute the transition formula (that is, to verify the rectangle rule), we would have to consider the behavior of each of the coefficients in the six regions depending on four possible cases of transition; that is, we would have to perform 24 computational procedures. To proceed with this task explicitly would put some strain on the reader and is not actually necessary in all instances. We will hence concentrate on a few dominant computational procedures and leave the remaining ones to the reader.

DEFINITION 3.1. Let $H_{T,U}=\{\bar{y}\}\subseteq Y$ be a vertex in Y. The *tableau* corresponding to $\bar{y}$ is the mapping

$$\Upsilon_{\bar{y}}:(T^c\cup U^c)\times(T\cup U\cup\{*\})\to\mathbb{R}$$

defined by $\Upsilon_{\bar{y}}(s,r)=\overline{\Upsilon}_{sr}(s\in T^c\times U^c,\ r\in T\times U\times\{*\})$, where $\overline{\Upsilon}$ is the $m\times(n+1)$ matrix



(3.1)

The entries of the matrix are defined as follows. The last column contains

$$\overline{\Theta}=-\mathfrak{A}_{-T}(\bar{y},\bar{\lambda})=-A_{-T}\bar{y}+\bar{\lambda}e_{-T} \tag{3.2}$$

(see Corollary 2.6) and the vector $\bar{y}_{-U}$ (i.e. the positive coordinates of $\bar{y}$). Next, $\bar{\gamma}_{\cdot}^{\cdot}$ is obtained via $\bar{\mu}^{i_0}=(\bar{\gamma}^{i_0},\bar{\nu}^{i_0})\in\mathbb{R}^{J-U}$ $(i_0\in T)$ and the requirement that

$$\mathfrak{A}_{\mathrm{T}}^{-U}\bar{\mu}^{i_0}=\begin{bmatrix}0\\ \vdots\\ 1\\ \vdots\\ 0\end{bmatrix}=e_{\mathrm{T}}^{i_0} \tag{3.3}$$

(cf. Lemma 2.1 and Definition 2.2). Similarly, $\bar{c}_{\cdot}^{\cdot}$ is obtained by inspecting (2.11), that is,

$$\bar{c}^{i_0}=-\mathfrak{A}_{-\mathrm{T}}^{-U}\bar{\mu}^{i_0}\in\mathbb{R}^{I-T}. \tag{3.4}$$

Finally, the quantities $\bar{d}^{\cdot}_{\cdot}$ and $\bar{\delta}^{\cdot}_{\cdot}$ have not been motivated as yet; the formal definition is given as follows. For $j_0 \in U$, vectors $\bar{\delta}^{j_0} \in \mathbb{R}^{J-U}$ and $\bar{\rho}^{j_0} = (\bar{\delta}^{j_0}, \bar{\sigma}^{j_0}) \in \mathbb{R}^{\mathbf{J}-U}$ are defined by the requirement that

$$\mathfrak{A}^{j_0}_{\mathbf{T}} = \mathfrak{A}^{-U}_{\mathbf{T}} \bar{\rho}^{j_0} \tag{3.5}$$

hold true. By nondegeneracy, $\bar{\rho}^{j_0}$ is indeed well defined (this is in fact the normal paradigm of changing the base in the LP case). Accordingly, for $j_0 \in U$ we define the vector $\bar{d}^{j_0} \in \mathbb{R}^{I-T}$ by

$$\bar{d}^{j_0} = A^{j_0}_{-T} - \mathfrak{A}^{-U}_{-\mathbf{T}} \bar{\rho}^{j_0}. \tag{3.6}$$

Remark 3.2. There is no harm in visualizing $\Upsilon_{\bar{y}}$ by $\overline{\Upsilon}$. However, with respect to a matrix the *ordering* of rows and columns sometimes is important. Thus, in a rigorous representation, $\Upsilon_{\bar{y}}$ is actually an *equivalence class* of matrices, to be obtained by permuting rows and columns of $\overline{\Upsilon}$ (including the row and column indices).

Given the definition of the tableau for $H_{T,U} = \{\hat{y}\}$, let us turn to the rectangle rule, which is a mapping of transforming general tableaus.

Fix $U \subseteq J$ and $T \subseteq I$. Let Υ be a mapping [the (T, U) *tableau*]

$$\Upsilon : (T^c \cup U^c) \times (T \cup \mathrm{U}) \to \mathbb{R},$$

and let

$$i_0 \in T, \qquad i_1 \in I - T.$$

Let

$$\hat{T} := T - i_0 + i_1,$$

and let $\hat{\Upsilon}$ denote a mapping [$(\hat{T}, U)$-*tableau*]

$$\hat{\Upsilon} : (\hat{T}^c \cup U^c) \times (\hat{T} \cup \mathrm{U}) \to \mathbb{R}.$$

The rectangle rule [for (i_0, i_1)] is a mapping that sends (T, U) tableaus into $(\hat{T}, U)$ tableaus, say

$$\mathscr{R}_{i_0, i_1} : \{\Upsilon\} \to \{\hat{\Upsilon}\}$$

as indicated by the familiar diagram shown in Figure 4.

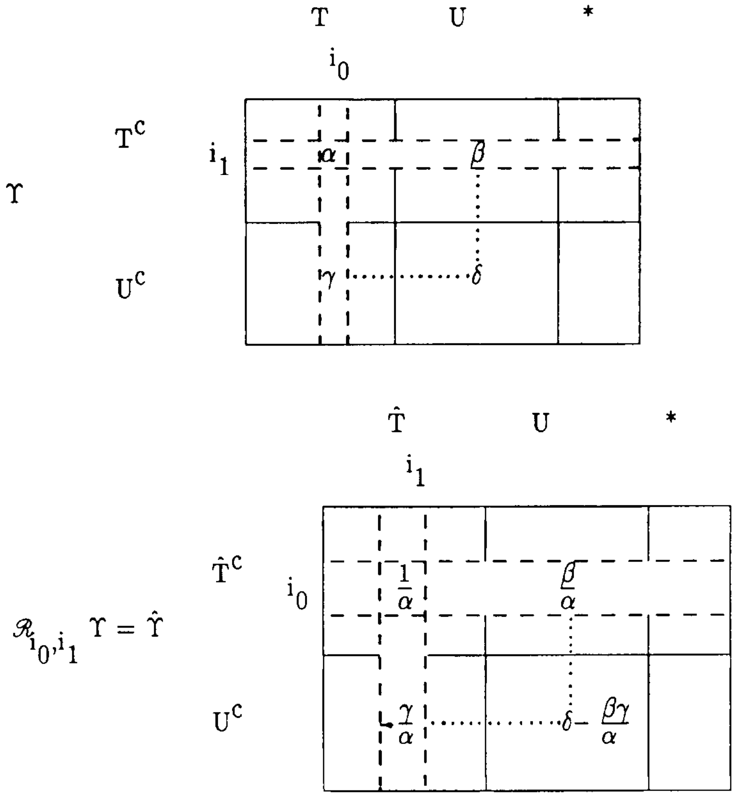


FIG. 4.

Of course, application of this kind of rectangle rule will correspond in case 1a to a transfer

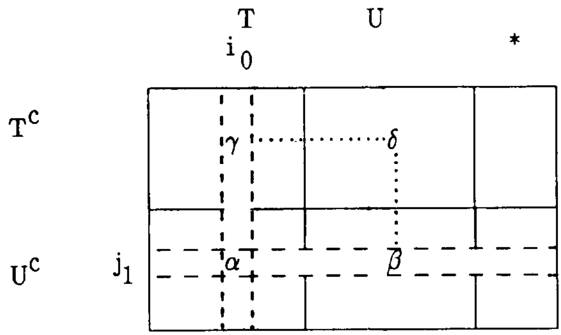
$$H_{T,U} \to H_{T-i_0+i_1,U} = H_{\hat{T},U}.$$

If we have to deal with a transfer

$$H_{T,U} \to H_{T-i_0,U+j_1} = H_{\hat{T},\hat{U}}$$

(corresponding to case 1b), then there is a corresponding $\mathscr{R}_{i_0,j_1}$. Here, $\mathscr{R}_{i_0,j_1}\Upsilon = \hat{\Upsilon}$ is a mapping as indicated via Figure 5.

The ordering of rows and columns is, of course, arbitrary—which is why a tableau perhaps is better thought of as a mapping. The fact that we have four



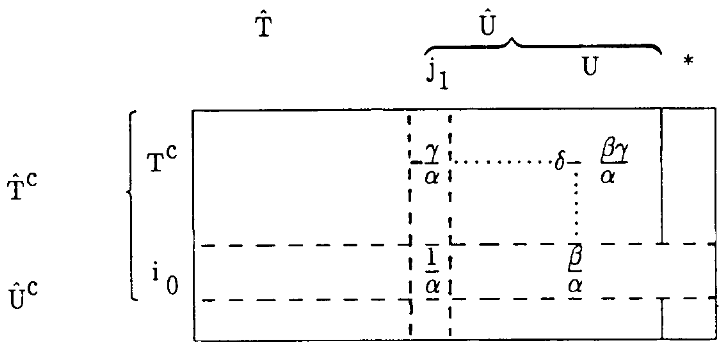


FIG. 5.

kinds of transitions (and hence four kinds of tableaus) must be taken into account repeatedly while implementing the algorithm.

THEOREM 3.3. *Let $H_{T,U} = \{\bar{y}\} \subseteq Y$ be a vertex in Y, and let $i_0 \in T$, $|T| \geqslant 2$. Suppose $\hat{y}$ is the vertex adjacent to $H_{T-i_0,U}$ other than $\bar{y}$, and assume that $\{\hat{y}\} = H_{T-i_0+i_1,U}$. Then the corresponding tableaus satisfy*

$$\Upsilon_{\hat{y}} = \mathscr{R}_{i_0 i_1} \Upsilon_{\bar{y}}. \tag{3.7}$$

PROOF. We have to compute the transition for six types of entries in $\Upsilon_{\hat{y}}$ and $\Upsilon_{\bar{y}}$. Let us first concentrate on the $U^c \times T$ block of $\Upsilon_{\bar{y}}$. The transitions should be

$$\gamma \to -\frac{\gamma}{\alpha} \qquad \text{for} \quad j_1 \in U^c, \quad i_0,$$

$$\delta \to \delta - \frac{\beta\gamma}{\alpha} \qquad \text{for} \quad j_1 \in U^c, \quad \hat{\imath}_0 \in T - i_0.$$

The entries of $\Upsilon_{\bar{y}}$ are

$$\gamma = \bar{\mu}_{j_1}^{i_0}, \quad \alpha = \bar{c}_{i_1}^{i_0}, \qquad j_1 \in U^c,$$

$$\beta = \bar{c}_{i_1}^{\hat{\imath}_0}, \quad \delta = \bar{\mu}_{j_1}^{\hat{\imath}_0}, \qquad \hat{\imath}_0 \in T - i_0.$$

Thus, the rectangle rule requires

$$\alpha \mapsto \frac{1}{\alpha} = \frac{1}{\bar{c}_{i_1}^{i_0}}, \qquad \beta \mapsto \frac{\beta}{\alpha} = \frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}},$$

$$\gamma \mapsto -\frac{\gamma}{\alpha} = -\frac{\bar{\mu}_{j_1}^{i_0}}{\bar{c}_{i_1}^{i_0}}, \qquad \delta \mapsto \delta - \frac{\beta\gamma}{\alpha} = \bar{\mu}_{j_1}^{\hat{\imath}_0} - \bar{\mu}_{j_1}^{i_0}\frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}}.$$

As for row j_1, consult Corollary 2.5. Clearly, (2.13) tells us that $-\gamma/\alpha$ is indeed the (j_1, i_1) entry of $\Upsilon_{\hat{y}}$, while (2.12) indicates that $\delta - \beta\gamma/\alpha$ is the $(j_1, \hat{\imath}_0)$ entry of $\Upsilon_{\hat{y}}$ (in the $U^c \times \hat{T}$ block).

The remaining computations, though sometimes tedious, are a mere formality. By virtue of our considerations in Section 2 we know that the tableau entries of $\Upsilon_{\bar{y}}$ determine $\hat{y}$. Now, as the entries of some $\Upsilon_.$ are defined formally by (3.1)–(3.6), we just have to verify the rectangle rule via the definitions (3.1)–(3.6). To this end, fix $\hat{\imath}_0 \in \hat{T}$, $\hat{\imath}_1 \in \hat{T}^c$. Also, denote the entries of $\Upsilon_{\hat{y}}$ with a $\hat{}$, e.g., $\hat{\mu}, \hat{\rho}$, etc. The same notation has been employed in Section 2.

First of all, let us take the $U^c \times U$ block, i.e., consider $\rho = (\delta, \sigma)$. As $\hat{\rho}^{j_0}$ (for $j_0 \in U = \hat{U}$) is defined via

$$\mathfrak{A}_{\hat{\mathrm{T}}}^{-U}\hat{\rho}^{j_0} = \mathfrak{A}_{\hat{\mathrm{T}}}^{j_0}, \tag{3.8}$$

we compute

$$\mathfrak{A}_{\hat{\mathrm{T}}}^{-U}\left(\bar{\rho}^{j_0} - \frac{d_{i_1}^{j_0}}{\bar{c}_{i_1}^{i_0}}\bar{\mu}^{i_0}\right). \tag{3.9}$$

The coordinates $i \in \hat{\mathbf{T}}$ of (3.9) are given by

$$\mathfrak{A}_i^{-U}\bar{\rho}^{j_0} + \frac{a_{i_1 j_0} - \mathfrak{A}_{i_1}^{-U}\bar{\rho}^{j_0}}{\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0}}\mathfrak{A}_i^{-U}\bar{\mu}^{i_0}$$

$$= \begin{cases} \mathfrak{A}_i^{j_0} + \dfrac{a_{i_1 j_0} - \mathfrak{A}_{i_1}^{-U}\bar{\rho}^{j_0}}{\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0}}\cdot 0 & \text{for} \quad i \neq i_1, \\[2ex] \mathfrak{A}_{i_1}^{-U}\bar{\rho}^{j_0} + \dfrac{a_{i_1 j_0} - \mathfrak{A}_{i_1}^{-U}\bar{\rho}^{j_0}}{\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0}}\mathfrak{A}_{i_1}^{-U}\bar{\mu}^{i_0} & \text{for} \quad i = i_1 \end{cases}$$

$$= \mathfrak{A}_i^{j_0},$$

that is, the coordinates are those of the right hand side of (3.8). Thus, the factor in parenthesis in (3.9) must be the left hand side of (3.8)—this takes care of the $\bar{\delta}_\cdot^\cdot$-entries in the $U^c \times U$ block.

Next, the $\bar{y}_\cdot$-entries, i.e. the $U^c \times \{*\}$ block, are obviously taken care of by Corollary 2.6, i.e., by (2.20).

We proceed with the $\bar{c}_\cdot^\cdot$-entries in the $T^c \times T$ block, using the fact that the rectangle rule has already been established for $\bar{\mu}$ versus $\hat{\mu}$. Hence, using the definition as provided in (3.4), we proceed as follows: First, for all $\hat{\imath}_0 \neq i_1$,

$$\hat{c}_{\hat{\imath}_1}^{\hat{\imath}_0} = -\mathfrak{A}_{\hat{\imath}_1}^{-U}\hat{\mu}^{\hat{\imath}_0} \qquad [\text{by } (3.4)]$$

$$= -\mathfrak{A}_{\hat{\imath}_1}^{-U}\left(\bar{\mu}^{\hat{\imath}_0} - \frac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}}\bar{\mu}^{i_0}\right) \qquad [\text{by Corollary 2.5, i.e., by } (2.12)]$$

$$= \begin{cases} \bar{c}_{\hat{\imath}_1}^{\hat{\imath}_0} - \dfrac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}}\bar{c}_{\hat{\imath}_1}^{i_0} & \text{for} \quad \hat{\imath}_1 \neq i_0 \quad [\text{by } (3.4)], \\[2ex] \dfrac{\bar{c}_{i_1}^{\hat{\imath}_0}}{\bar{c}_{i_1}^{i_0}} & \text{for} \quad \hat{\imath}_1 = i_0 \quad [\text{by } (3.3)]. \end{cases} \tag{3.10}$$

Similarly, for $\hat{\imath}_0 = i_1$,

$$\hat{c}^{i_1}_{\hat{\imath}_1} = -\mathfrak{A}^{-U}_{\hat{\imath}_1}\hat{\mu}^{i_1}$$

$$= -\mathfrak{A}^{-U}_{\hat{\imath}_1}\left(-\frac{\overline{\mu}^{i_0}}{\overline{c}^{i_0}_{i_1}}\right)$$

$$= \begin{cases} -\dfrac{\overline{c}^{i_0}_{\hat{\imath}_1}}{\overline{c}^{i_0}_{i_1}} & \text{for} \quad \hat{\imath}_1 \neq i_0 \quad [\text{by } (3.4)], \\[2ex] \dfrac{1}{\overline{c}^{i_0}_{i_1}} & \text{for} \quad \hat{\imath}_1 = i_0 \quad [\text{by } (3.3)]. \end{cases} \tag{3.11}$$

Obviously, (3.10) and (3.11) establish the rectangle rule for the $T^c \times T$ block. As for the $d^{\cdot}_{\cdot}$ in the $T^c \times U$ block, we have by (3.3.6)

$$\hat{d}^{\hat{\jmath}_0}_{\hat{\imath}_1} = a_{\hat{\jmath}_0\hat{\imath}_1} - \mathfrak{A}^{-U}_{\hat{\imath}_1}\left(\overline{\rho}^{\hat{\jmath}_0} - \frac{\overline{d}^{\hat{\jmath}_0}_{i_1}}{\overline{c}^{i_0}_{i_1}}\overline{\mu}^{i_0}\right)$$

$$= \begin{cases} \overline{d}^{\hat{\jmath}_0}_{\hat{\imath}_1} - \dfrac{\overline{c}^{i_0}_{\hat{\imath}_1}\overline{d}^{\hat{\jmath}_0}_{i_1}}{\overline{c}^{i_0}_{i_1}} & \text{for} \quad \hat{\imath}_1 \neq i_0, \\[2ex] \dfrac{\overline{d}^{\hat{\jmath}_0}_{i_1}}{\overline{c}^{i_0}_{i_1}} & \text{for} \quad \hat{\imath}_1 = i_0, \end{cases} \tag{3.12}$$

using (3.6), (3.4), (3.3), and (3.5).

Finally, the $\overline{\Theta}_{\cdot}$ in the $T^c \times \{*\}$ blocks are transformed by using (3.2); thus, for $\hat{\imath}_1 \neq i_0$,

$$\hat{\Theta}_{\hat{\imath}_1} = -\mathfrak{A}_{\hat{\imath}_1\cdot}(\hat{y}, \hat{\lambda})$$

$$= \overline{\Theta}_{\hat{\imath}_1} + \mathfrak{A}^{-U}_{\hat{\imath}_1\cdot}\frac{\overline{\Theta}_{i_1}}{\overline{c}^{i_0}_{i_1}}\overline{\mu}^{i_0} \qquad [\text{by Corollary 2.6}]$$

$$= \overline{\Theta}_{\hat{\imath}_1} - \frac{\overline{\Theta}_{i_1}\overline{c}^{i_0}_{\hat{\imath}_1}}{\overline{c}^{i_0}_{i_1}} \qquad [\text{by } (3.4)], \tag{3.13}$$

and for $\hat{\imath}_1 = i_0$,

$$\hat{\Theta}_{\hat{\imath}_1} = -\mathfrak{A}_{i_0\cdot}(\hat{y},\hat{\lambda})$$

$$= -\underbrace{\mathfrak{A}_{i_0\cdot}(\bar{y},\bar{\lambda})}_{0\text{ [by Corollary 2.6]}} + \frac{\bar{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}}\mathfrak{A}_{i_0\cdot}^{-U}\bar{\mu}^{i_0}$$

$$= \frac{\bar{\Theta}_{i_1}}{\bar{c}_{i_1}^{i_0}} \quad \text{[by (3.3)]}. \qquad \blacksquare \quad (3.14)$$

The further development is rather straightforward. There are four kinds of possible transitions $H_{T,U} \to H_{\hat{T},\hat{U}}$ when passing from one vertex to an adjacent one via some edge. To each of these transitions, there corresponds a rectangle rule—we have explicitly indicated two of them. Now we have

THEOREM 3.4. *Let $H_{T,U} = \{\bar{y}\}$ and $H_{\hat{T},\hat{U}} = \{\hat{y}\}$ be adjacent vertices. Suppose $\Upsilon_{\bar{y}}$ is the tableau corresponding to $\bar{y}$. Then $\Upsilon_{\hat{y}}$ is obtained by the rectangle rule (i.e., the one corresponding to $H_{T,U} \to H_{\hat{T},\hat{U}}$).*

PROOF. We shall discuss briefly all four cases 1a–2b. Now, 1a has already been dealt with. For 1b, we return to the presentation exhibited in Sections 2.4 and 2.5; here we have to replace (2.9) by

$$\bar{\theta}^{i_0} = \frac{\bar{y}_{j_1}}{\bar{\gamma}_{j_1}^{i_0}}, \qquad (3.15)$$

thus assuming that a transition

$$H_{T,U} \to H_{T-i_0,U+j_1}$$

takes place. Again we compute $\hat{y} = y^{\bar{\theta}^{i_0}}$.

In doing so, we realize that the quantities of the tableau $\tau_{\bar{y}}$ are sufficient in order to perform all necessary computations. Hence, it suffices to again check the rectangle rule ($\mathscr{R}_{i_0,j_1}$, that is) for case 1b. This amounts to juggling the quantities specified by (3.2)–(3.6). As the details are to be perceived by close analogy to the treatment of 1a, we shall not offer a further discussion.

As to cases 2a, b, we abbreviate the discussion—in principle we have to introduce another canonical parametrization. Consider the vertex

$$\{\bar{y}\} = H_{T,U},$$

and let $\bar{\lambda} = A_{i\cdot}\bar{y}$ $(i \in T)$. Pick $j_0 \in U$; it follows from nondegeneracy that

$$L_T^{j_0 - U} = \left\{\rho = (\delta, \sigma) \in \mathbb{R}^{-(U-j_0)} \times \mathbb{R} \,\middle|\, \mathfrak{A}_{\mathbf{T}}^{-(U-j_0)}(\delta, \sigma) = 0\right\} \tag{3.16}$$

is a linear subspace of $\mathbb{R}^{-U} \times \mathbb{R}$ of dimension 1.

Again in view of the nondegeneracy, it is clear that Equation (3.16), i.e.

$$\mathfrak{A}_{\mathbf{T}}^{j_0} = \mathfrak{A}_{\mathbf{T}}^{-U} \bar{\rho}_{-j_0}^{j_0}$$

defines the vector $\bar{\rho}^{j_0}$ uniquely, and the mapping

$$\begin{aligned} &\theta \to (\bar{y}, \bar{\lambda}) - \theta\left(\bar{\rho}^{j_0} \oplus O_{U-j_0}\right), \\ &\mathbb{R} \to \mathbb{R}^n \times \mathbb{R} \end{aligned} \tag{3.17}$$

defines the canonical parametrization of the affine subspace

$$(\bar{y}, \bar{\lambda}) + \left(L_T^{j_0 - U} \oplus O_{U-j_0}\right).$$

Of course, the projection

$$\theta \to y^\theta = \bar{y} - \theta\left(\bar{\delta}^{j_0} \oplus O_{U-j_0}\right)$$

also parametrizes an affine subspace of $\mathbb{R}^n$; this latter one contains $H_{T,U-j_0}$ (and $H_{T,U}$). Thus, the analogue of Theorem 2.4 is as follows:

There is $\bar{\theta}^{j_0} > 0$ *such that*

$$H_{T,U-j_0} = \left\{y^\theta \,\middle|\, 0 \leqslant \theta \leqslant \bar{\theta}^{j_0}\right\}. \tag{3.18}$$

$\bar{\theta}^{j_0}$ *is explicitly computed by*

$$\bar{\theta}^{j_0} = \min\left\{ \frac{\bar{\lambda} - A_{i\cdot}\bar{y}}{\bar{\sigma}^{j_0} - A_{i\cdot}^{-U}\bar{\delta}^{j_0}} \,\middle|\, i \in T^c,\ \bar{\sigma}^{j_0} > A_{i\cdot}^{-U}\bar{\delta}^{j_0} \right\}$$

$$\wedge \min\left\{ \frac{\bar{y}_j}{\bar{\delta}_j^{j_0}} \,\middle|\, j \in U^c,\ \bar{\delta}_j^{j_0} > 0 \right\}. \tag{3.19}$$

From this vantage point the reader now views the scene that has so extensively been described in case 1. We will leave him there to his own efforts—if necessary. ■

In this section, our final task is to consider briefly the *initial tableau*. This turns out to be a nice and simple shape.

THEOREM 3.5. *Let* $j_1 \in J$ *and* $\bar{y} = e^{j_1} = (0,\dots,0,1,0,\dots,0) \in Y$. *Suppose* $i_0 \in I$ *is such that*

$$a_{i_0 j_1} = \max_{i \in I} a_{i j_1}.$$

Then $\{e^{j_1}\} = H_{T,U}$, *with* $T = \{i_0\}$ *and* $U = J - \{j_1\}$, *is a vertex in* Y, *and the corresponding tableau is given by* $\Upsilon_{e^{j_1}}$, *which is indicated by the following matrix*:

$$\begin{array}{ll|c|ccc|c|}
 & & T & & U & & \\
 & & i_0 & & j & & * \\
\hline
 & & -1 & & \vdots & & \vdots \\
 & & \vdots & & \vdots & & \vdots \\
T^c & i & -1 & \cdots\cdots & \bar{d}_i^j & \cdots & \bar{\Theta}_i \\
 & & \vdots & & \vdots & & \vdots \\
 & & -1 & & \vdots & & \vdots \\
\hline
U^c & j_1 & 0 & 1\cdots\cdots & 1 & \cdots 1 \cdots & 1 \\
\hline
\end{array} \tag{3.20}$$

Here

$$\overline{\Theta}_i = a_{i_0 j_1} - a_{i j_1} \qquad (i \in T^c) \tag{3.21}$$

and

$$\overline{d}_i^j = a_{ij} - a_{ij_1} + a_{i_0 j_1} - a_{i_0 j} \qquad (i \in T^c, \quad j \in U). \tag{3.22}$$

PROOF. Clearly, $\{e^{j_1}\} = H_{T,U}$ is a vertex. Note that

$$\overline{\lambda} = a_{i_0 j_1}$$

holds true. All we have to do is verify the entries of the matrix using (3.2)–(3.6). In view of (3.2) we have

$$\overline{\Theta}_i = \overline{\lambda} - A_{i\cdot} e^{j_1} = a_{i_0 j_1} - a_{i j_1},$$

which shows (3.21). Next, exploit (3.5) in order to obtain

$$\begin{aligned}
\overline{\rho}^j &= \left(\mathfrak{A}_{\mathbf{T}}^{-U}\right)^{-1} \mathfrak{A}_{\mathbf{T}}^{j} \\
&= \begin{bmatrix} a_{i_0 j_1} & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} a_{i_0 j} \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ -1 & a_{i_0 j_1} \end{bmatrix} \begin{bmatrix} a_{i_0 j} \\ 1 \end{bmatrix} \\
&= \left(1, a_{i_0 j_1} - a_{i_0 j}\right).
\end{aligned}$$

The first coordinate of $\overline{\rho}^j$ is $\overline{\delta}_{j_1}^j$, which equals 1.

Next, (3.6) leads to

$$\begin{aligned}
\overline{d}_i^j &= a_{ij} - \mathfrak{A}_i^{-U} \overline{\rho}^j \\
&= a_{ij} - \left(a_{ij_1}, -1\right)\left(1, a_{i_0 j_1} - a_{i_0 j}\right)
\end{aligned}$$

and thus (3.22).

The remaining computations, easy as they are, will not be carried out explicitly. ■

4. IMPLEMENTING THE ALGORITHM

Suppose that, starting out from some vertex $\{\bar{y}\} = H_{T,U}$, we have left K_{i_0} ($i_0 \in T$); hence a transition takes places along the edge $H_{T-i_0,U}$ and case 1a or 1b will prevail. From Theorem 2.4 and the following presentation we know that this depends essentially on the minimizing argument that yields $\bar{\theta}^{i_0}$ in Section 2.4.2. Clearly, the quantities competing for this minimizer are basically available in the tableau $\Upsilon_{\bar{y}}$. For, in solving the definition presented by (4.2) and (4.4) below according to Section 2.4.2, it turns out that we have to look for the minimizer that yields the expression

$$\min\left\{ \frac{\overline{\Theta}_i}{\bar{c}_i^{i_0}} \,\middle|\, i \in T^c,\ \bar{c}_i^{i_0} > 0 \right\} \wedge \min\left\{ \frac{\bar{y}_j}{\bar{\gamma}_j^{i_0}} \,\middle|\, j \in U^c,\ \bar{\gamma}_j^{i_0} > 0 \right\}.$$

Verbally, this means that we take the quotients of column $*$ and column i_0 of $\Upsilon_{\bar{y}}$ coordinatewise and look for the minimizing row. According as this yields some $i_1 \in T^c$ or some $j_1 \in U^c$, we end up with 1a or 1b. Note that the quotient minimizing row is unique by nondegeneracy.

It is not hard to prove the generalization of this.

THEOREM 4.1. *Let $\{\bar{y}\} = H_{T,U}$ be a vertex in Y with tableau $\Upsilon_{\bar{y}}$. Denote the last column of $\Upsilon_{\bar{y}}$ by $\tau_{\cdot *}$ $[=(\overline{\Theta}, \bar{y}_{-U})]$. Let $H_{T',U'}$ be an adjacent edge (so $T' = T - \rho$ or $U' = U - \rho$), and let $\tau_{\cdot\rho}$ be the ρth column of $\Upsilon_{\bar{y}}$. Next, let σ be a row of $\Upsilon_{\bar{y}}$ (i.e. $\sigma \in T^c$ or $\sigma \in U^c$) such that*

$$\frac{\tau_{\sigma *}}{\tau_{\sigma\rho}} = \min\left\{ \frac{\tau_{\sigma' *}}{\tau_{\sigma'\rho}} \,\middle|\, \tau_{\sigma'\rho} > 0,\ \sigma' \in \{\text{rows of } \Upsilon_{\bar{y}}\} \right\}. \tag{4.1}$$

Then the following hold true:

(1) *σ is uniquely determined.*

(2) *$\{\hat{y}\} = H_{\hat{T},\hat{U}}$ with $\hat{T} = T' + \sigma$ or $\hat{U} = U' + \sigma$ (chosen appropriately) is the vertex other than $\bar{y}$ that is adjacent to $H_{T',U'}$.*

(3) *$\Upsilon_{\hat{y}} = \mathscr{R}_{\sigma\rho}\Upsilon_{\bar{y}}$.*

PROOF. Obvious. ■

Finally, we have to ponder the terminating condition. To this end, consider the version of the LH algorithm discussed in [11, Chapter 1, Section 1, Theorem 1.14], which is based on the set

$$\mathscr{S}_n = \left\{ x \in X \middle| x_i > 0,\, y \in K_i\ (i \in I);\ y_j > 0,\, x \in L_j\ (j \in J - n) \right\}.$$

Geometrically, this means that the starting vertex in Y is e^n and that the first $H_{T,U}$ is some $H_{i_0, J-n}$. Now, obviously the process terminates once either n is added to the indices in U so as to constitute $H_{T,U}$ with $n \in U$, or n is removed from R such that $G_{R,V}$ satisfies $n \notin R$.

In any case, the algorithm terminates once the index n appears afresh the first time. If we complete the rectangle rule, then the equilibrium coordinates can be simply read from the tableau, as they are listed in the last row.

Concluding, the implementing procedure for the modified LH algorithm is described as follows: Given matrices A and B, perform the following steps.

Step INITIALIZE.

1. Choose $n_0 \in J$ arbitrarily.
2. Choose $i_0 \in I$ such that

$$a_{i_0 n_0} = \max_i a_{i n_0}. \tag{4.2}$$

3. Choose $j_0 \in J$ such that

$$b_{i_0 j_0} = \max b_{i_0 j}. \tag{4.3}$$

If $j_0 = n_0$, then STOP. (e^{i_0}, e^{j_0}) is a (pure) equilibrium point. Otherwise, set up

Step INITIAL TABLEAUS. The initial tableau arising from the matrix A is uniquely described by (3.20), (3.21), (3.22). This defines $\Upsilon_{\bar{y}}$ with $\bar{y} = e^{n_0}$. The initial tableau arising from the matrix B is obtained by exchanging B^T and A, J and I, n and m, etc. That is, we have

$I - i_0$

		j_0	i	$\square$
		-1	⋮	⋮
$J - j_0$	j	-1	⋯ $\bar{\bar{d}}_j^i$ ⋯	⋯ $\bar{\bar{\Theta}}_j$
		⋮	⋮	⋮
		-1	⋮	⋮
	i_0	0	$1 \quad 1 \cdots 1$	1

(4.4)

Here

$$\overline{\overline{\Theta}}_j = b^T_{j_0 i_0} - b^T_{j i_0}$$

$$= b_{i_0 j_0} - b_{i_0 j} \tag{4.5}$$

and

$$\overline{\overline{d}}^i_j = b^T_{ji} - b^T_{j i_0} + b^T_{j_0 i_0} - b^T_{j_0 i}$$

$$= b_{ij} - b_{i_0 j} + b_{i_0 j_0} - b_{i j_0}. \tag{4.6}$$

Having thus established the initial tableaus, CONTINUE *with the algorithm.*

Step CONTINUE. Having obtained the information j_0 form the B-tableau, determine i_1 (or j_1) to be the minimizer of the (well-defined) quotients of column $*$ and column j_0 in the A-tableau, i.e.,

$$\min_{\sigma'} \frac{\tau_{\sigma' *}}{\tau_{\sigma' j_0}} = \frac{\tau_{\sigma *}}{\tau_{\sigma j_0}}.$$

Traditionally, σ is called the *pivot*. Apply the rectangle rule, say $\mathscr{R}_{j_0 i_1}$, to the A-tableau. CONTINUE with the B-side. Generally, the information contained in an index ρ (the pivot) from the previous side determines a column in the tableau of the present side. The minimizer σ of the quotients of the last column and column ρ is the next pivot. It determines the rectangle rule $\mathscr{R}_{\rho\sigma}$ to be applied to the present tableau. Also, the pivot σ is the information to be used at the next step with the tableau of the other side. As far as the pivot satisfies $\sigma \neq n_0 \in J$, CONTINUE with this step; otherwise move to TERMINATE.

Step TERMINATE. If the pivot satisfies $\sigma = n_0 \in J$, the algorithm STOPS (after the last $\mathscr{R}_{\rho n_0}$ has been performed). The A-tableau as depicted in (3.1) contains the positive coordinates $j \in U^c$ of $\bar{y}$, i.e., the vector $\bar{y}_{-U} = \bar{y}_{U^c}$, in the $U^c \times \{*\}$ block. Correspondingly, the B-tableau contains some $\bar{x}_{V^c}$ in the corresponding block. Augmenting these quantities by an appropriate string of zeros yields an equilibrium point $(\bar{x}, \bar{y})$.

Flowcharts of the APL program are given in Figures 6–8. A listing is given in Table 1.

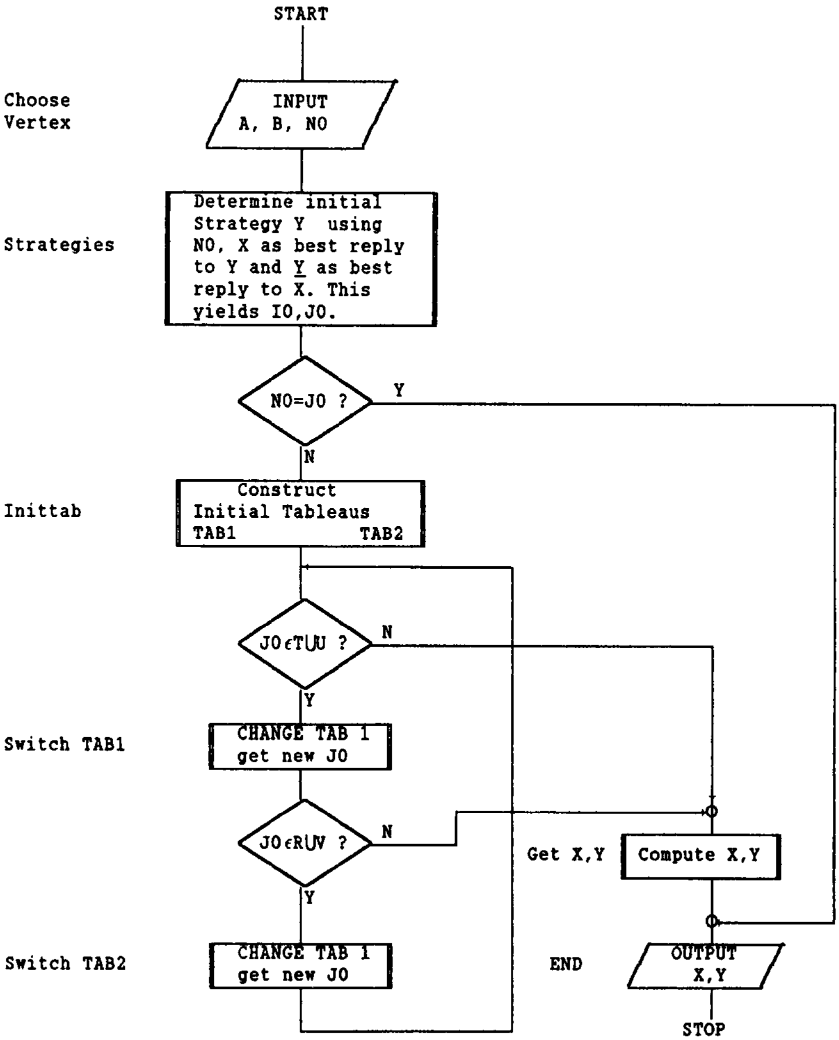


FIG. 6. Program LH.

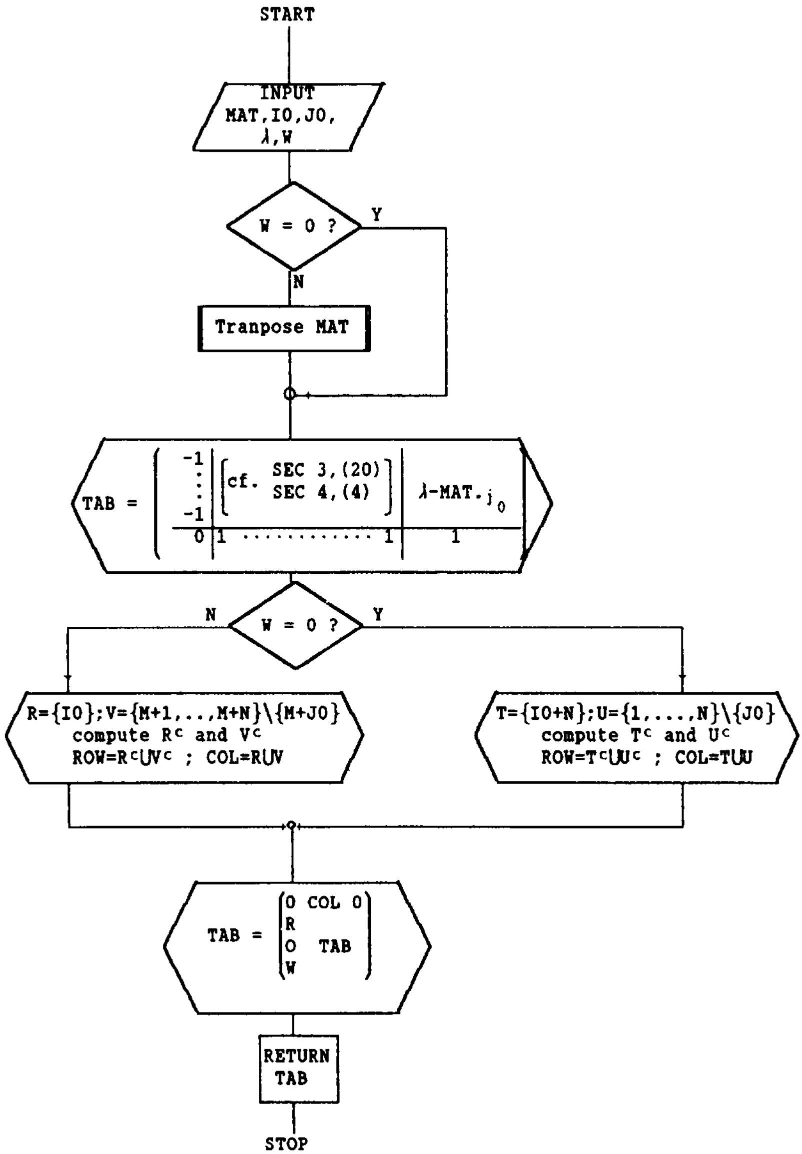


FIG. 7. Subprogram Inittab.

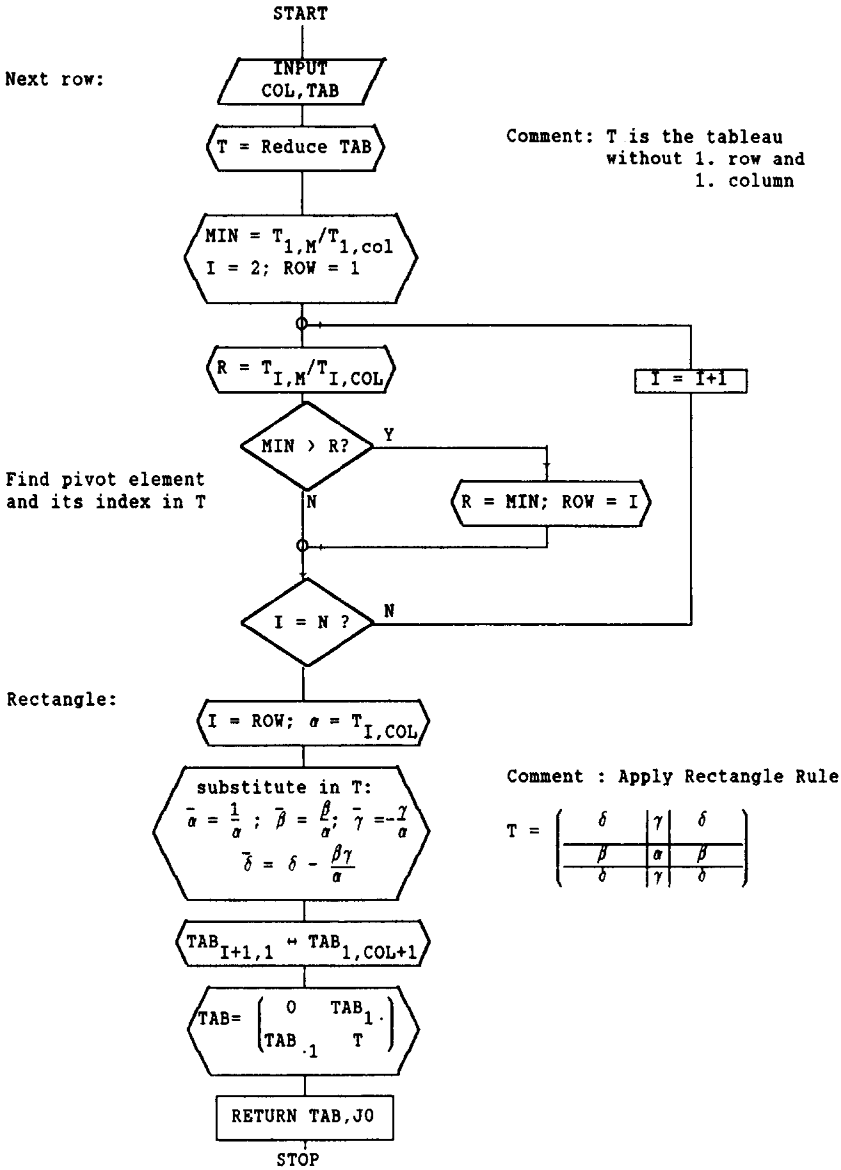


FIG. 8. Subprogram Switchtab. The subprogram is divided into two subprograms: (1) for given column (COL) compute the row for the pivot element; (2) change the tableau.

TABLE I

```
      ∇ R←A LH B;M;N;C;NO;X;Y;IO;LAY;JO;LAX;TAB1;TAB2;ROW;COL;ZEA;ZEB
[1]     M←(⍴A)[1] ⍝                            Get dimension of A.
[2]     N←(⍴A)[2]
[3]    CHOOSEVERTEX: ⍝                          Select a start strategy for player 1 or 2.
[4]     'Select a column for player 2 out of 1,...,',⍕N
[5]     '               or for player 1 out of ',(⍕N+1),',...,',⍕M+N
[6]     NO←⎕ ⋄ C←0 ⍝                           Read input to NO.C=0 means player 2.
[7]     →(~NO∊⍳N+M)/CHOOSEVERTEX ⍝             Is input correct?
[8]     →(NO≤N)/STRATEGIES ⍝                   Is player 1 selected?
[9]    TRANSPOSE: ⍝                             Then change roles of player 1 and 2,
[10]    C←⍉A ⋄ A←⍉B ⋄ B←C ⍝                    transpose matrices and
[11]    C←N ⋄ N←M ⋄ M←C ⍝                      exchange M and N.
[12]    NO←NO-M ⋄ C←1 ⍝                        C=1 means player 1 was selected.
[13]   STRATEGIES: ⍝                            NO, IO, JO are the indices of 1 in strategies
[14]    X←M⍴0 ⋄ Y←N⍴0 ⍝                        Y , X ,Y̲. Find IO, JO, such that
[15]    IO←A[;NO]⍳LAY←⌈/A[;NO] ⍝               X is best reply to Y
[16]    X[IO]←Y[NO]←1 ⍝                        and Y̲ is best reply to X.
[17]    JO←B[IO;]⍳LAX←⌈/B[IO;] ⍝               If JO=NO, meaning Y=Y̲, then (X,Y)
[18]    →(JO=NO)/END ⍝                         is an EqP with payoff (LAX,LAY). Then STOP.
[19]   INITTABLEAUS: ⍝                          Else construct the initial tableaus for
[20]    TAB1←A INITTAB IO,NO,LAY,0 ⍝           player 1
[21]    TAB2←B INITTAB JO,IO,LAX,1 ⍝           and player 2 (1 means transpose B).
[22]   SWITCHTAB1: ⍝                            JO was the number of the last used row.
[23]    →(N<COL←¯1+TAB1[1;]⍳JO)/GETXY ⍝        If there is a corresponding column in TAB1
[24]    ROW←TAB1 NEXTROW COL ⍝                 then find the row index of the pivot element
[25]    TAB1←(ROW,COL)RECTANGLE TAB1 ⍝         in TAB1 to change TAB1. RECTANGLE calculates the
[26]   SWITCHTAB2: ⍝                            new JO (global in CHANGE) for TAB2. If there
[27]    →(M<COL←¯1+TAB2[1;]⍳JO)/GETXY ⍝        is no corresponding column to JO, and EqP
[28]    ROW←TAB2 NEXTROW COL ⍝                 has been reached.
[29]    TAB2←(ROW,COL)RECTANGLE TAB2 ⍝         If TAB2 was changed correctly
[30]    →SWITCHTAB1 ⍝                          then consider TAB1 again.
```

TABLE 1. *Continued.*

```
[31]  GETXY: ⍝                              Calculate strategies X and Y from
[32]   X←Mρ0 ⍝                              the last columns of TAB1 and TAB2.
[33]   X[((ZEB>N)/ZEB)-N]←((ZEB←1↓TAB2[;1])>N)/1↓TAB2[;M+2]
[34]   Y←Nρ0
[35]   Y[((0<ZEA)∧ZEA≤N)/ZEA]←((0<ZEA)∧(ZEA←1↓TAB1[;1])≤N)/1↓TAB1[;N+2]
[36]  END: ⍝                                If player 1 was selected, Y,X is the
[37]   →(C=0)/ANSWER ⋄ C←X ⋄ X←Y ⋄ Y←C ⍝ result, else X,Y.
[38]  ANSWER: 'Equilibrium point found with strategies : ',(⌽X),' for player 1'
[39]   '                                      and ',(⌽Y),' for player 2'
[40]   R←X,Y
    ∇

    ∇ ROW←TAB NEXTROW COL;TA;R;MIN;M
[1]    M←(ρTAB)[2] ⋄ ROW←0 ⍝                If no pivot element can be found return 0.
[2]    →(0=∨/TA←⎕CT<1↓TAB[;COL+1])/END ⍝ Eliminate zeros in COL column,
[3]    R←(TA/1↓TAB[;M)÷TA/1↓TAB[;COL+1] ⍝divide M+1 column by COL column and
[4]    →((L/θ)=MIN←L/(R>0)/R)/END ⍝          find the smallest value, if it exits,
[5]    ROW←(TA\R)⍳MIN ⍝                      and its index.
[6]   END: ⍝                                 Return pivot row.
    ∇

    ∇ R←MAT INITTAB PARAM;IO;JO;LAMBDA;W;DELTA;GAMMA;C;D;TETA;Y;ROW;COL;V;U;N;M
[1]    IO←PARAM[1] ⋄ JO←PARAM[2] ⋄ LAMBDA←PARAM[3] ⋄ W←PARAM[4] ⋄ →(W=0)/M1
[2]   ⍝ PARAM contains IO, JO, the no. of pos. coordinates in strategies of players 1 and 2,
[3]   ⍝ resp, the payoff for this strategy from matrix MAT, and the boolean variable W that
[4]   ⍝ indicates, whether the matrix MAT has to be transposed (W=1).
[5]    MAT←⍉MAT
[6]   M1:M←(ρMAT)[1] ⋄ N←(ρMAT)[2] ⍝MAT is an M×N matrix.
[7]    V←Mρ1 ⋄ V[IO]←0 ⍝              Take the inverse strategies to reduce the matrix.
[8]    U←Nρ1 ⋄ U[JO]←0
[9]    C←((M-1),1)ρ¯1 ⍝               Now calculate the components of the tableau :-------
```

TABLE 1. *Continued.*

```
[10]   D←MAT[IO;JO]+(U/V⌿MAT)-(⍉((N-1),M-1)⍴V/MAT[;JO])+((M-1),N-1)⍴U/MAT[IO;] ⍝      |
[11]   TETA←LAMBDA-V/MAT[;JO] ⍝      (  c       d        teta)                      |
[12]   GAMMA←0 ⍝                     (gamma   delta       y)                        |
[13]   DELTA←(1,N-1)⍴1 ⍝                                                            |
[14]   Y←1 ⍝                                                                        |
[15]   R←(C,D,TETA),[1]GAMMA,DELTA,Y ⍝ and order them to the tableau -------------------
[16]   →(W=0)/M2 ⍝                   Calculate the column and row vectors for player 1 or 2:
[17]   ROW←((⍳(IO-1)),IO↓⍳M),JO+M ⍝  RC={1,...,M}\{IO}          VC={JO+M}
[18]   COL←IO,M+(⍳(JO-1)),JO↓⍳N ⍝    R={IO}                     V={M+1,...,M+N}\{JO+M}
[19]   →M3
[20]  M2:ROW←(N+(⍳(IO-1)),IO↓⍳M),JO ⍝ TC={N+1,...,N+M}\{N+IO}  UC={JO}
[21]   COL←(N+IO),(⍳(JO-1)),JO↓⍳N ⍝ T={N+IO}                    U={1,...,N}\{JO}
[22]  M3:R←(0,COL,0),[1]ROW,R
     ∇
     ∇ NTAB←PIVOT RECTANGLE TAB;M;N;ROW;COL;RO;CO;PI;J1
[1]    M←¯1+(⍴TAB)[1] ⋄ N←¯1+(⍴TAB)[2] ⍝ Get the dimension of the tableau
[2]    ROW←PIVOT[1] ⋄ COL←PIVOT[2] ⍝     and the index of the pivot element.
[3]    NTAB←1 1↓TAB ⍝                    Tableau without row and column vector.
[4]    RO←M⍴1 ⋄ RO[ROW]←0 ⍝              Take all rows, except the pivot row,
[5]    CO←N⍴1 ⋄ CO[COL]←0 ⍝              all columns, except the pivot column.
[6]    NTAB←RO⍀CO\(NTAB←CO/RO⌿NTAB)-((RO⌿NTAB[;,COL])+.×(CO/NTAB[,ROW;]))÷PI←NTAB[ROW;COL]
[7]    NTAB[;COL]←-1↓TAB[;COL+1]÷PI ⍝    Calculate the outer elements (see line 6),
[8]    NTAB[ROW;]←1↓TAB[ROW+1;]÷PI ⍝     the pivot column (see line 7) and pivot row.
[9]    NTAB[ROW;COL]←1÷PI ⍝              Replace the pivot element.
[10]   J1←TAB[1;COL+1]⋄JO←TAB[ROW+1;1] ⍝Take row and column (global JO),
[11]   NTAB←TAB[1;],[1](1↓TAB[;1]),NTAB ⍝copy old row and column vectors to new
[12]   NTAB[ROW+1;1]←J1 ⍝                tableau and exchange row and column.
[13]   NTAB[1;COL+1]←JO ⍝                JO will be used by LH.
     ∇
```

REFERENCES

1 V. Aggarval, On the generation of all equilibrium points for bimatrix games through the Lemke-Howson algorithm, *Math. Programming* 4:233–234 (1973).

2 M. Bastian, Another note on bimatrix games, *Math. Programming* 11:299–300 (1976).

3 P. E. M. Borm, On refinements in bimatrix games, Preprint, Dept. of Mathematics, Univ. of Nijmegen, 1989.

4 P. Borm, A. Gijsberts, and S. Tijs, A geometric-combinatorial approach to bimatrix games, *Methods Oper. Res.* 59 (1988).

5 B. C. Eaves, The linear complementary problem, *Management Sci.* 17:612–634 (1971).

6 C. E. Lemke, Bimatrix equilibrium points and mathematical programming, *Management Sci.* 7:681–689 (1965).

7 C. E. Lemke and J. T. Howson, Jr., Equilibrium points in bimatrix games, *SIAM J. Appl. Math.* 12:413–423 (1964).

8 T. Parthasarathy and T. E. S. Raghavan, *Some Topics in Two-Person Games*, Modern Anal. and Comput. Methods Sci. and Math. 22, American Elsevier, 1971.

9 L. S. Shapley, A note on the Lemke-Howson algorithm, *Math. Programming Study* 1:175–189 (1974).

10 J. Rosenmüller, On a generalization of the Lemke-Howson algorithm to noncooperative n-person games, *SIAM J. Appl. Math.* 21:73–79 (1971).

11 J. Rosenmüller, *The Theory of Games and Markets*, North Holland, 1981.

12 M. Todd, Comments on a note by Aggarval, *Math. Programming* 10:130–133 (1976).

13 M. Todd, Bimatrix games—an addendum, *Math. Programming* 14:112–115 (1978).

14 R. Wilson, Computing equilibria of n-person games, *SIAM J. Appl. Math.* 21:80–87 (1971).

15 P. Wolfe, The simplex algorithm for quadratic programming, *Econometrica* 27:382–398 (1959).